A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Miqueias Sousa dos Santos |
| *Student Number* | 2016287 |
| *Module Title* | *HDip in AI Applications* |
| *Assessment Title* | *CA1 In Class Assessment* |
| *Assessment Due Date* | *28/04/2024* |
| *Date of Submission* | *27/04/2024* |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Table of Contents

1. Introduction1
2. Motivation1
3. Description of the Problem Domain1
4. Data characterization and pre-processing1
5. Hyperparameter tuning in machine learning3
6. Interpretation and Explanation of Results3
7. Overfitting/Underfitting/Generalization3
8. Rationale for Chosen Models4
9. Visualizations to Support Findings……………………………………………………4
10. Conclusion6

References7

1. **Introduction**

The housing market is a complex ecosystem influenced by various socio-economic factors, policies, and trends. Accurately predicting housing prices is crucial for stakeholders like buyers, sellers, and policymakers. Leveraging machine learning algorithms can comprehensively analyze housing data for informed predictions. This project aims to develop and deploy machine learning models for predicting housing prices based on relevant features.

(96 words)

1. **Motivation**

The project is motivated by the housing market's significant role in the economy and society. Housing is essential and a major financial asset. Precise price predictions can empower decision-making for buyers, sellers, real estate agents, and policymakers. By using advanced machine learning techniques, the project aims to enhance prediction accuracy and improve decision-making in the housing market.

(82 words)

1. **Description of the Problem Domain**

The project focuses on understanding the factors influencing housing prices and building models to predict them accurately. Variables such as location, property size, amenities, neighbourhood characteristics, economic indicators, and market trends affect housing prices. Analysing these factors requires sophisticated tools like machine learning. Additionally, factors like population growth, urban development, interest rates, and regulatory changes also play significant roles. By integrating these diverse factors into our models, we aim to provide robust and reliable predictions that can assist stakeholders in making informed decisions in the dynamic housing market landscape.

(141 words)

1. **Data characterization and pre-processing**

In the process of preparing the data for analysis, I began by ensuring that the dataset adhered to the specific requirements outlined by California regulations, which involved tailoring it to contain 300 rows. Following this, I meticulously examined the data for any missing values and handled them by removing them from the dataset to maintain data integrity.

Given that my focus was primarily on numerical data for predictive modelling, I made the decision to exclude the "ocean\_proximity" column, which likely contained categorical data. This streamlined the dataset, ensuring it comprised only numerical features essential for the analysis. A screenshot of a computer

Description automatically generated

To gain deeper insights into the distribution of the dataset and understand the range and spread of the numerical features, I visualized the data using histograms. This visualization technique allowed me to discern any patterns or anomalies within the dataset, providing valuable context for the predictive modeling efforts.

A group of blue and white graphs

Description automatically generated

Moving forward, I split the dataset into features (X) and the target variable (y), with X representing the independent variables used for prediction and y representing the dependent variable, specifically, the median house value.

To better understand the relationships between the features and the target variable, I employed a correlation heatmap. This visualization tool allowed me to identify potential predictors strongly correlated with the median house value, aiding in the selection of the most influential features for the predictive models.

A screenshot of a computer

Description automatically generated

Lastly, I specified various test sizes for cross-validation, including 20%, 25%, and 30%. These test sizes determined the proportion of the dataset reserved for evaluating the model's performance across different splits, enabling me to assess the variation in accuracy effectively.

By meticulously undertaking these steps in the data preparation phase, I ensured that the dataset was well-prepared for subsequent predictive modeling tasks, laying a solid foundation for robust model evaluation and reliable predictive outcomes.

1. **Hyperparameter tuning in machine learning**

The primary purpose of hyperparameter tuning in machine learning is to find the optimal set of hyperparameters that produces the most accurate predictions for a given model. Hyperparameters are the configuration settings used to train a machine learning model, and they can significantly influence the performance of the model.

There are several techniques for hyperparameter tuning, including: **Grid Search (GridSearchCV)**: This is a traditional way of performing hyperparameter tuning. It works by defining a grid of hyperparameters and then evaluating the model performance for each point on the grid. You can then choose the point that gives the best performance. While this can be very effective, it can also be very time-consuming, especially if you have many hyperparameters.

Below is the GridSearchCV for hyperparameter tuning with a Random Forest Regressor used in the project.

A screenshot of a computer code

Description automatically generated

**Random Search**: This method randomly selects combinations of hyperparameters to train the model and evaluate performance. The main advantage of this method is that it can be more efficient than grid search if you have a large number of hyperparameters.

(450 words)

1. **Interpretation and Explanation of Results**

**Linear Regression, Ridge, and Lasso:** These linear models exhibit decent performance, capturing some of the variability in housing prices. However, they may struggle to capture more complex relationships present in the data, as indicated by their moderate RMSE and R2 scores.

**Support Vector Regression (SVR):** SVR performs poorly compared to other algorithms, with negative R2 scores suggesting poor generalization to unseen data. This indicates that SVR may be underfitting the data and failing to capture the underlying patterns adequately.

**Random Forest Regressor and Gradient Boosting Regressor:** These ensemble methods outperform the linear models, demonstrating lower RMSE and higher R2 scores. They excel in capturing complex relationships in the data, leading to improved predictive performance.

**Neural Network (MLPRegressor):** The Neural Network model shows the worst performance among all algorithms, with high RMSE values and negative R2 scores. This suggests that the model may be overfitting to the training data and failing to generalize well to new instances.

(101 words)

1. **Overfitting/Underfitting/Generalization**

**Overfitting:** Overfitting occurs when a model learns the training data too well, capturing noise and outliers, which leads to poor performance on unseen data. The Neural Network model exhibits signs of overfitting, as indicated by its high training score and poor performance on test data.

**Underfitting:** Underfitting occurs when a model is too simple to capture the underlying patterns in the data, resulting in poor performance on both training and test data. SVR demonstrates signs of underfitting, as it fails to capture the relationships present in the data, leading to negative R2 scores.

**Generalization:** Generalization refers to a model's ability to perform well on unseen data. Random Forest Regressor and Gradient Boosting Regressor demonstrate good generalization, as they perform well on both training and test data, indicating that they capture the underlying patterns in the data without overfitting.

(100 words)

1. **Rationale for Chosen Models**

The Random Forest Regressor and Gradient Boosting Regressor were strategically selected based on their capacity to navigate intricate data relationships and deliver robust predictive outcomes. These ensemble methods integrate numerous decision trees, offering a potent advantage in mitigating overfitting while enhancing generalization capabilities.

By harnessing the collective intelligence of multiple trees, these models excel in capturing nuanced patterns within the dataset, thereby elevating their predictive accuracy and reliability. This ensemble approach not only bolsters performance but also instills resilience against data noise and variability, rendering Random Forest and Gradient Boosting Regressors as optimal choices for the complex predictive landscape of housing price estimation.

(100 words)

1. **Visualizations to Support Findings**

A screenshot of a computer

Description automatically generated

**A group of text on a white background

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A white background with black text

Description automatically generated**

1. **Conclusion**

In conclusion, ensemble methods such as the Random Forest Regressor and Gradient Boosting Regressor emerge as superior performers when tasked with predicting housing prices. Demonstrating superior generalization capabilities and adeptness at capturing complex data relationships, these models outshine both linear models and neural networks. However, it's crucial to approach model selection and tuning with vigilance, particularly considering the potential pitfalls of overfitting and underfitting in the dynamic landscape of the housing market domain. By striking a balance between model complexity and interpretability, practitioners can harness the full potential of ensemble methods to make informed decisions and navigate the intricacies of housing price prediction with confidence.

(105 words)

**References**

Chen, T. and Guestrin, C. (2016). ‘XGBoost: A Scalable Tree Boosting System’, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794.

California Department of Real Estate. (n.d.). Real Estate Law and Regulations. Available at: https://dre.ca.gov/Licensees/LicenseLaw.html (Accessed: 27 April 2024).

Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O’Reilly Media.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

McKinney, W. and others. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O’Reilly Media.

OpenAI. (n.d.). ChatGPT. Available at: https://openai.com/research/chatgpt (Accessed: 27 April 2024).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … and Vanderplas, J. (2011). ‘Scikit-learn: Machine Learning in Python’, Journal of Machine Learning Research, 12, pp. 2825-2830.

Raschka, S. and Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing.

Scikit-learn. (n.d.). Machine Learning in Python. Available at: https://scikit-learn.org/stable/ (Accessed: 27 April 2024).

Seabold, S. and Perktold, J. (2010). ‘Statsmodels: Econometric and Statistical Modeling with Python’, Proceedings of the 9th Python in Science Conference, pp. 92-96.

TensorFlow. (n.d.). An Open Source Machine Learning Framework for Everyone. Available at: https://www.tensorflow.org/ (Accessed: 27 April 2024).